A 26-Contact Tangible Pen-Like Input Device for Capacitive Displays

Lisa A Elkin, Jean-Baptiste Beau, Géry Casiez, Daniel Vogel

A 26-Contact Tangible Pen-Like Input Device for Capacitive Displays

Lisa A. Elkin
Paul G. Allen School of Computer Science & Engineering,
University of Washington,
Seattle, United States

School of Computer Science,
University of Waterloo,
Waterloo, Canada
lelkin@cs.washington.edu

Géry Casiez
Univ. Lille, UMR 9189 - CRIStAL,
Lille, France

Inria, France

Institut Universitaire de France (IUF)

School of Computer Science,
University of Waterloo,
Waterloo, Canada
gery.casiez@univ-lille.fr

Jean-Baptiste Beau
School of Computer Science,
University of Waterloo,
Waterloo, Canada
jalbeau@uwaterloo.ca

Daniel Vogel
School of Computer Science,
University of Waterloo,
Waterloo, Canada
dvogel@uwaterloo.ca

CHI '20 Extended Abstracts, April 25–30, 2020, Honolulu, HI, USA.

ACM ISBN 978-1-4503-6819-3/20/04.
DOI: https://doi.org/10.1145/3334480.3383157

Abstract
We designed and created a new self-contained tangible pen-like input device prototype that can sense all 26 contacts and works with any capacitive display using a conductive case designed with pliable corners. Contacts are distinguished using the device angle from an internal IMU. We further designed a 3D "mirror" visualization that displays a re-configurable mapping of commands to contacts to enable discovery of command-to-contact mappings.

Author Keywords
Pen input; tangible interfaces; command selection.

CCS Concepts
•Human-centered computing → Graphics input devices;

Introduction
There is a class of tangible input devices that not only provide positional input, but also detect which geometric feature contacts a surface. This class has an advantage over traditional input methods because it integrates command selection (how it contacts) and parameter manipulation (positional movement). A common example is a digital stylus which uses a nib end and an eraser end to switch between drawing and erasing. Extending this approach has been

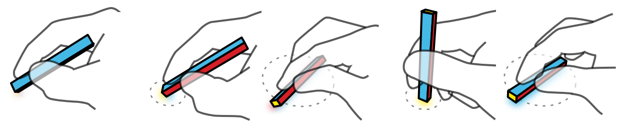

Figure 1: Conté, a tangible pen-like input device (from [4]).

demonstrated in research like ToolStone [3], and Vogel and Casiez's pen-like input device, called Conté [4].

Conté is held in one hand, and manipulated so that one of the 8 corners, 12 edges, or 6 sides of its cuboid shape contacts a tablet (Figure 1). In theory, these 26 different contacts enable selecting among 26 different commands. However, Vogel and Casiez's prototype only sensed 10 different contacts and required a diffuse illumination table top.

We designed and created a new self-contained Conté prototype that can sense all 26 contacts and works with any capacitive display using a conductive case designed with pliable corners. Contacts are distinguished using the device angle from an internal IMU, and a 3D "mirror" visualization displays a re-configurable mapping of commands to contacts to enable discovery of command-to-contact mappings.

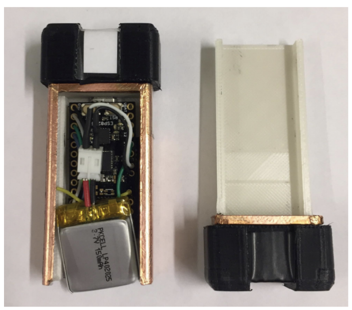

Figure 2: Our prototype accommodates a microcontroller, IMU, and battery. The main case is printed in PLA wrapped in copper tape, and the ends are printed with Conductive 95 Shore A TPU.

26-Contact Conte Device

Our 26-contact capacitive Conté device (Figure 2) retains the cuboid shape of Vogel and Casiez's simple 10-command prototype, but its design and implementation are different in all other aspects. Our device is 85 $\times$ 30 $\times$ 15 mm to accommodate a microcontroller, IMU, and battery. In our companion paper [1], we show this size actually has advantages over the smaller size used by Vogel and Casiez. Unlike the previous device, ours senses 26 contacts on a conventional capacitive touch surface, the first device of its kind to do so. It is not a polished consumer product, but with routine recalibration, it is effective at demonstrating the potential of the hardware approach.

Case Design

The main case is printed in PLA wrapped in copper tape, and the ends are printed with *Conductive 95 Shore A Thermoplastic Polyurethane (TPU)* [1]. The end corners are designed with hollow pockets to make them pliable enough to register on standard capacitive sensors, but stiff enough to feel like corners. The narrow sides of each end project 1.5mm forming 11mm wide valleys. When combined with the narrower main case, this creates patterns of one, two, or four capacitive touches when corners, edges, or sides are contacting the display. White marks are added to one end to create visual asymmetry, so different orientations are distinguishable.

Internal Hardware

Orientation, acceleration, and magnetic field data from the 9 DOF MPU9250 IMU are processed by a dedicated sensor-fusion chip at 100 Hz, then passed to an Arduino-compatible ESP8285 microcontroller which packages the data, then sends it as a UDP stream over on-board WIFI at 100 Hz. All internal hardware is powered by a 3.7V, 150mAh LiPo battery.

Software

A host application, written in Java with Processing[2], combines the UDP stream of IMU data with a TUIO stream of touch events from a standard touch input device to create an OSC stream of device events[3]. The events include `down`, `move`, and `up` events to describe which corner, edge, or side is contacting the display, the (x, y) position(s) of the touch

[1] http://rubber3dprinting.com/pi-etpu-95-250-carbon-black/
[2] https://processing.org
[3] TUIO (www.tuio.org) and OSC (opensoundcontrol.org)

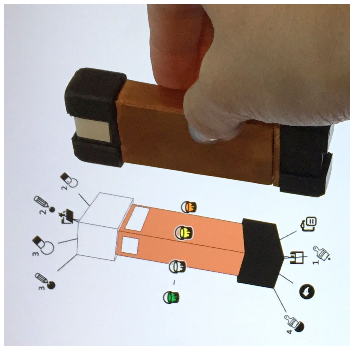

Figure 3: Conté on-screen guide showing command-to-contact mapping. Note how pattern of white marks creates 3D landmarks.

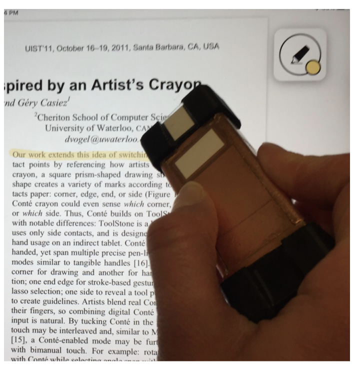

Figure 4: Conté being used in an iOS iPad PDF application

points, and raw IMU data for custom usage. In the case of two or four touch points (most edges and sides), a single down and up event is sent when the first or last touch point is registered. In addition, an `air` event sends IMU data when the device is not contacting the touchscreen.

Generating these events requires touch filtering and contact classification.

When the device is dragged quickly, the reported touchscreen events can “skip” with spurious up and down events, leaving gaps in what should be a continuous line. To counteract this, up and down touchscreen events are ignored when they occur within 300ms and within a distance threshold (46mm for single points, 15mm for two or more points). An exception is made when more than 75% of the device acceleration is along the normal vector of the display, then the anti-skip filter is ignored and the touchscreen up event is immediately accepted. All values were determined by trial and error, to minimize spurious events with reasonable latency.

The contact is classified from IMU data using a decision tree. Training data was gathered by touching the device to a touchscreen 40 times per contact, each time at a different orientation over the possible range. Three frames of IMU readings were recorded after each touch down. This was performed by the first author, since this is hardware calibration to determine the relative orientation of the IMU to the touchscreen. The resulting 3,120 data points were used to train a decision tree in Weka with 10-fold cross-validation, resulting in 99.2% accuracy. During actual usage, the host application uses the trained decision tree to classify the three IMU data frames after a `down` event, using a majority voting scheme with the last frame acting as tie breaker. Note IMU “drift” and magnetic interference reduce this ideal accuracy during real usage, requiring users to perform a drift-reset after about 4 to 5 minutes of intense usage. Even with this precaution, there are still some misclassifications, most often with adjacent contacts, like a corner and short end edge. Professional-grade engineering and IMU hardware would reduce the need for manual drift correction substantially.

Using a dedicated host to generate events means applications can run on any platform. We demonstrate Conté with an iPad in the accompanying video and we conducted an experiment using a Windows touchscreen laptop in our companion paper [1].

“Mirror” Visualization Guide

To enable discovery of command-to-contact mappings, there is an on-screen guide showing a mirrored view of the device as a 3D rendering, with non-occluded contacts labelled with command icons (Figure 3). Pilot studies indicated this was easier to use than a non-mirrored guide.

In our demonstration prototype, this visualization is activated after 500ms by holding down the ‘m’ key on an external keyboard, and releasing ‘m’ dismisses it. In the future, other activation methods could be used like squeezing the sides, tapping the end with a finger when the device is not contacting the display, or perhaps even holding the device still above the screen for a moment. The short delay functions to separate novice guidance from expert usage like many previous novice-to-expert interaction techniques (e.g. [2]), and this short time penalty may even encourage memorization of command mappings. In a formal experiment in our companion paper [1], we show that almost all 26 contacts can be learned in a two-hour session. Participants using our Conté device with the mirror visualization had an average of 94% recall after 24 hours.

Application Examples

Annotating a PDF requires precise input commonly performed with a stylus and frequent command-selection. It is a natural application of Conté. Corners can be used as pens or highlighter; small and medium edges to scroll the page in different directions and zoom in and out; long edges can be used as a ruler; and the ends can be used to stamp dates and signatures on documents (Figure 4).

Some other possible applications include a physical control for a video player where laying Conté on different sides can execute commands like play and pause, and moving Conté quickly to the left or right can rewind and fast-forward; a game controller that can interact with on-screen content as a joystick or more directly by being placed on the screen and manipulated; and of course, as a digital artist's crayon.

Conclusion

Vogel and Casiez's original Conté paper was about the *idea* of a pen-like tangible input device [4]. It demonstrated how transitioning to different contacts enabled a range of interactions like a multipurpose pen, an interactive ruler, a temporary mouse, and more. However, their prototype only sensed 10-contacts and it required a specialized tabletop. We designed and implemented a new version of Conté which realizes their vision of a self-contained hardware device that detects all 26 contacts and works with standard touch displays. Our re-imagined device is a significant improvement and shows the full potential of using tangible pen-like input.

Acknowledgements

This work made possible by NSERC Discovery Grant 2018-05187, the Canada Foundation for Innovation Infrastructure Fund 33151 "Facility for Fully Interactive Physio-digital Spaces," and Ontario Early Researcher Award ER16-12-184.

REFERENCES

[1] Lisa A. Elkin, Jean-Baptiste Beau, Géry Casiez, and Daniel Vogel. 2020. Manipulation, Learning, and Recall with Tangible Pen-Like Input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '20)*. ACM, New York, NY, USA. DOI: http://dx.doi.org/10.1145/3313831.3376772

[2] Gordon Kurtenbach and William Buxton. 1994. User Learning and Performance with Marking Menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '94)*. ACM, New York, NY, USA, 258–264. DOI: http://dx.doi.org/10.1145/191666.191759

[3] Jun Rekimoto and Eduardo Sciammarella. 2000. ToolStone: Effective Use of the Physical Manipulation Vocabularies of Input Devices. In *Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology (UIST '00)*. ACM, New York, NY, USA, 109–117. DOI: http://dx.doi.org/10.1145/354401.354421

[4] Daniel Vogel and Géry Casiez. 2011. Conté: Multimodal Input Inspired by an Artist's Crayon. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST '11)*. ACM, New York, NY, USA, 357–366. DOI: http://dx.doi.org/10.1145/2047196.2047242